



# Towards a new generation of parton densities with deep learning models

Stefano Carrazza<sup>a</sup>, Juan Cruz-Martinez<sup>✉</sup>

TIF Lab, Dipartimento di Fisica, Università degli Studi di Milano and INFN Sezione di Milano, Via Celoria 16, 20133 Milan, Italy

Received: 15 July 2019 / Accepted: 1 August 2019  
© The Author(s) 2019

**Abstract** We present a new regression model for the determination of parton distribution functions (PDF) using techniques inspired from deep learning projects. In the context of the NNPDF methodology, we implement a new efficient computing framework based on graph generated models for PDF parametrization and gradient descent optimization. The best model configuration is derived from a robust cross-validation mechanism through a hyperparametrization tune procedure. We show that results provided by this new framework outperforms the current state-of-the-art PDF fitting methodology in terms of best model selection and computational resources usage.

## 1 Introduction

In perturbative QCD, parton distribution functions (PDFs) are used to describe the non-perturbative structure of hadrons [1–3]. These functions are typically determined by means of a supervised regression model which compares a wide set of experimental data with theoretical predictions computed with a PDF parametrization. A truthful determination of PDFs and its uncertainties are important requirements when producing theoretical prediction for precision studies in high energy physics. From a methodological point of view, the choice of a regression model and its uncertainty treatment is a crucial decision which will impact the quality of PDFs and its theoretical predictions.

The aim of this paper is to describe a new regression strategy framework inspired on deep learning techniques for the NNPDF methodology [4]. The NNPDF methodology uses machine learning techniques in combination of Monte Carlo data generation to extract PDFs from experimental data. The NNPDF approach was pioneer in using artificial neural networks for the PDF parametrization and genetic algorithms for optimization. The NNPDF fitting framework has been con-

stantly reviewed and upgraded in the last years, by including new features and methodological improvements which enhanced the quality of the released PDF sets [5–7]. Motivated by the new technologies and algorithms in use by the machine learning community, we dedicate this study to assess the impact of such new strategies in a modern PDF determination.

We focus our study on three issues. The first consists in improving performance of the current NNPDF approach, where each PDF replica fit requires a large number of CPU hours to complete, e.g. in a global PDF determination a single fit takes  $\mathcal{O}(30)$  CPU hours. The second regards the efficiency (or lack thereof) of neural network optimization through genetic algorithms. Finally we aim to achieve a flexible framework in order to easily change neural network models and tune its architecture and learning strategy.

The paper is organized as follows. In Sect. 2 we summarize briefly the current NNPDF methodology, highlighting the main differences with respect to the new approach proposed in this paper as well as the testing setup we use for benchmarking and tuning the results. In Sect. 3, we introduce the hyperparametrization procedure adopted to find the optimal learning strategy. Finally, in Sect. 4 we show as example some preliminary fits using this new technology.

## 2 Methodology

### 2.1 The NNPDF methodology

The NNPDF collaboration implements by default the Monte Carlo approach to PDF fits. The goal of such strategy is to provide an unbiased determination of PDFs with reliable uncertainty. The NNPDF methodology is based on the Monte Carlo treatment of experimental data, the parametrization of PDFs with artificial neural networks, and the minimization strategy based on genetic algorithms.

<sup>a</sup> e-mail: stefano.carrazza@cern.ch

In the next paragraphs we outline the most relevant aspects of the NNPDF3.1 methodology. An exhaustive overview is beyond the scope of this paper, we invite the reader to review [4, 8] for further details.

The Monte Carlo approach to experimental data consists in generating artificial data replicas based on the experimental covariance matrix of each experiment. This procedure allows to propagate experimental uncertainties into the PDF model by performing a PDF fit for each data replica. Usually, PDF sets generated from such approach are composed by 100–1000 replicas.

The experimental data used in the PDF fit is preprocessed according to a cross-validation strategy based on randomly splitting the data for each replica into a training set and a validation set. The optimization is then performed on the training set while the validation loss is monitored and used as stopping criterion to reduce overlearning.

In the NNPDF fits, PDFs are parameterized at a reference scale  $Q_0$  and expressed in terms of a set of neural networks corresponding to a set of basis functions. Each of these neural networks consists of a fixed-size feedforward multi-layer perceptron with architecture 1-2-5-3-1. The input node ( $x$ ) is split by the first layer on the pair  $(x, \log(x))$ . The two hidden layers (of 5 and 3 nodes) use the sigmoid activation function while the output node is linear:

$$f_i(x, Q_0) = A_i x^{-\alpha_i} (1-x)^{\beta_i} \text{NN}_i(x), \quad (1)$$

where  $\text{NN}_i$  is the neural network corresponding to a given flavour  $i$ , usually expressed in terms of the PDF evolution basis  $\{g, \Sigma, V, V_3, V_8, T_3, T_8, c^+\}$ .  $A_i$  is an overall normalization constant which enforces sum rules and  $x^{-\alpha_i} (1-x)^{\beta_i}$  is a preprocessing factor which controls the PDF behaviour at small and large  $x$ . In order to guarantee unbiased results, in the current NNPDF methodology both the  $\alpha_i$  and  $\beta_i$  parameters are randomly selected within a defined range for each replica at the beginning of the fit and kept constant thereafter.

Unlike usual regression problems, where during the optimization procedure the model is compared directly to the training input data, in PDF fits the theoretical predictions are constructed through the convolution operation per data point between a fastkernel table (FK) as presented in Refs. [9, 10], which encodes the theoretical computation, and the PDF model, following the process type of the data point. For DIS-like processes the convolution is performed once, while for hadronic-like processes PDFs are convoluted twice.

The optimization procedure consists in minimizing the loss function

$$\chi^2 = \sum_{i,j}^{N_{\text{dat}}} (D - P)_i \sigma_{ij}^{-1} (D - P)_j, \quad (2)$$

where  $D_i$  is the  $i$ -nth artificial data point from the training set,  $P_i$  is the convolution product between the fastkernel tables for point  $i$  and the PDF model, and  $\sigma_{ij}$  is the covariance matrix between data points  $i$  and  $j$  following the  $t_0$  prescription defined in appendix of Ref. [11]. This covariance matrix can include both experimental and theoretical components as presented in Ref. [12].

Concerning the optimization procedure, so far, only genetic algorithms (GA) have been tuned and used. In summary the procedure consists in initializing the weights of the neural network for each PDF flavour using a random gaussian distribution and checking that sum rules are satisfied. From that first network 80 mutant copies are created based on a mutation probability and size to update the weights. The training procedure is fixed to 30k iterations and stopping is determined using a simple look-back algorithm which stores the best weights for the lowest validation loss value.

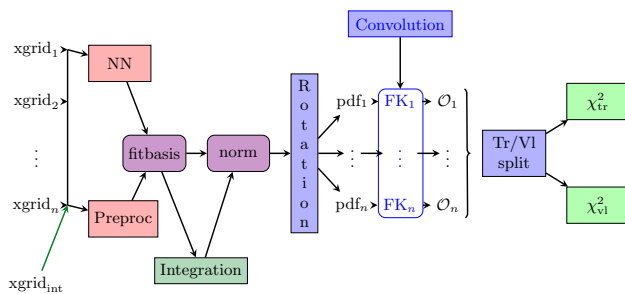
## 2.2 A new methodological approach

The methodology presented above is currently implemented in a C++ code, introduced for the first time in official releases in NNPDF3.0 [8] and which relies on a very small set of external libraries. This feature can become a shortcoming as the monolithic structure of the codebase greatly complicates the study of novel architectures and the introduction of modern machine learning techniques developed during the last few years.

Our target in this work is to construct a new framework that to allow for the enhancement of the methodology. In order to achieve our goal we rebuild the code using an object oriented approach that will allow us to modify and study each bit of the methodology separately.

We implement the NNPDF regression model from scratch in a python-based framework in which every piece aims to be completely independent. We choose Keras [13] and TensorFlow [14] in order to provide the neural network capabilities for the framework as they are some of the most used and well documented libraries, sometimes also used in the context of PDFs [15, 16]. In addition, the code design abstracts any dependence on these libraries in order to be able to easily implement other machine learning oriented technologies in the future. This new framework, by making every piece subjected to change, opens the door to a plethora of new studies which were out of reach before.

For all fits shown in this paper we utilize gradient descent (GD) methods to substitute the previously used genetic algorithm. This change can be shown to greatly reduce the computing cost of a fit while maintaining a very similar (and in occasions improved)  $\chi^2$ -goodness. The less stochastic nature of GD methods also produces more stable fits than its GA counterparts. The main reason why the GD methods had not been tested before were due to the difficulty of computing



**Fig. 1** Diagrammatic view of the `n3fit` code. Each different block is set as a different layer, following a structure similar to Keras. The red squared blocks correspond to blocks with fittable parameters

the gradient of the loss function (mainly due to the convolution with the fastkernel tables) in a efficient way. This is one example on how the usage of new technologies can facilitate new studies thanks to differentiable programming and distributed computing.

We also use just one single densely connected network as opposed to a separate network for each flavour. As previously done, we fix the first layer to split the input  $x$  into the pair  $(x, \log(x))$ . We also fix 8 output nodes (one per flavour) with linear activation functions. Connecting all different PDFs we can directly study cross-correlation between the different PDFs not captured by the previous methodology.

As we change both the optimizer and the architecture of the network, it is not immediately obvious which would be the best choice of parameters for the NN (which are collectively known as hyperparameters). Thus, we implement in this framework the hyperopt library [17] which allow us to systematically scan over many different combinations of hyperparameters finding the optimal configuration for the neural network. We detail the hyperparameter scan in Sect. 3.

### 2.3 A new fitting framework: `n3fit`

In Fig. 1 we show a schematic view of the full new methodology which we will refer to from now on as `n3fit`. The  $xgrid_1 \dots xgrid_n$  are vectors containing the  $x$ -inputs of the neural network for each of the experiments entering the fit. These values of  $x$  are used to compute both the value of the NN and the preprocessing factor, thus computing the unnormalized PDF. The normalization values  $A_i$  are then computed at every step of the fitting (using the  $xgrid_{int}$  vector as input), updating the “norm” layer and producing the corresponding normalized PDF of Eq. (1).

Before obtaining a physical PDF we apply a basis rotation from the fit basis,  $\{g, \Sigma, V, V_3, V_8, T_3, T_8, c^+\}$ , to the physical one, namely,  $\{\bar{s}, \bar{u}, \bar{d}, g, d, u, s, c(\bar{c})\}$ . After this procedure we have everything necessary to compute the value of the PDF for any flavour at the reference scale  $Q_0$ .

All fittable parameters live in the two red blocks, the first named NN [by default a neural network composed by densely connected layers corresponding to the NN of Eq. (1)] and the second the preprocessing  $\alpha$  and  $\beta$  which are free to vary during the fit (in NNPDF3.1 for each replica  $\alpha_i$  and  $\beta_i$  are fixed during the fit). In the next, when we refer to the neural network parameters we will be referring collectively to the parameters of these two blocks.

As in this new methodology each block is completely independent we can swap them at any point, allowing us to study how the different choices affect the quality of the fit. All the hyperparameters of the framework are also abstracted and exposed (crucial for the study shown in Sect. 3). It also allow us to study many different architectures unexplored until now in the field of PDF determination.

The PDFs, as seen in Sect. 2.1, cannot be compared directly to data, therefore it is necessary to bring the prediction of the network (the  $pdf_i$  of Fig. 1) to a state in which it can be compared to experimental data. For that we need to compute the convolution of the PDFs with the fastkernel tables discussed in Sect. 2.1 which produces a bunch of observables  $O_1 \dots O_n$  with which we can compute the loss function of Eq. (2).

The first step of the convolution is to generate a rank-4 luminosity tensor (for DIS-like scenarios this tensor is equivalent to the PDF):

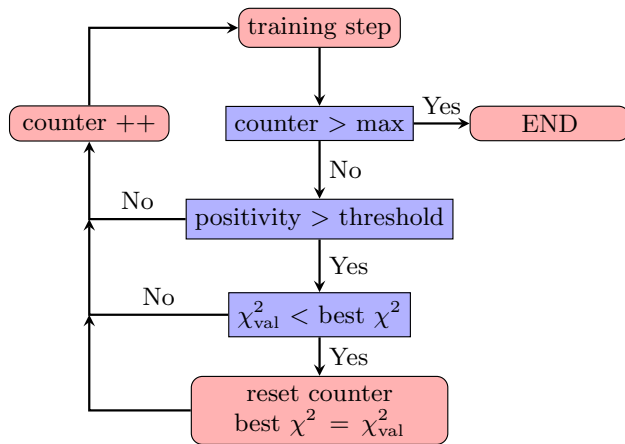
$$\mathcal{L}_{i\alpha j\beta} = f_{i\alpha} f_{j\beta}, \quad (3)$$

where the latin letters refer to flavour index while the greek characters refer to the index on the respective grids on  $x$ . The observable is then computed by contracting the luminosity tensor with the rank-5 fastkernel table for each separate dataset.

$$O^n = FK_{i\alpha j\beta}^n \mathcal{L}_{i\alpha j\beta}, \quad (4)$$

where  $n$  corresponds to the index of the experimental data point within the dataset. The computation of the observables is the most computationally expensive piece of the fit and the optimization and enhancement of this operation will be the object of future studies.

Before updating the parameters of the network we split the output into a training and validation set (selected randomly for each replica) and monitor the value of  $\chi^2$  for each one of these sets. The training set is used for updating the parameters of the network while the validation set is only used for early stopping. The stopping algorithm is presented in Fig. 2. We then train the network until the validation stops improving. From that point onwards, and to avoid false positives, we enable a patience algorithm which waits for a number of iterations before actually considering the fit finished.



**Fig. 2** Flowchart describing the patience algorithm of the `n3fit` code. The positivity constraint becomes more and more restrictive as the fit advances

A last block to review is the positivity constraint in Fig. 2. We only accept points for stopping for which the PDF is known to produce positive predictions for special set of pseudo data which tests the predictions for multiple processes in different kinematic ranges ( $x$ ,  $Q^2$ ). This mechanism follows closely that used in previous versions of NNPDF [4, 8].

The loss function defined in Eq. (2) is minimized in order to obtain the best set of parameters for the NN. We restrict ourselves to the family of Gradient Descent algorithms with adaptive moment where the learning rate of the weights is dynamically modified. In particular we focus on Adadelta [18], Adam [19] and RMSprop [20]. These three optimizer follow a similar gradient descent strategy but differ on the prescription for weight update.

## 2.4 Environment setup: data and theory

Benchmarking and validation of the new approach is done using as baseline the setup for NNPDF3.1 NNLO [4]. This means we will be using the same datasets and cuts, together with the same fraction of validation data for cross-validation although the stopping criteria is different (Fig. 2). This setup is named “global”, as it includes all datasets used in NNPDF3.1 NNLO with 4285 data points.

We also define, in order to facilitate the process of benchmarking and validation, a reduced dataset with only DIS-type data with 3092 data points. Namely, all datasets from the “global” setup that are not hadronic. We call this setup “DIS”. This reduced setup has a main advantage: in a DIS-like process there is only one PDF involved, this simplifies enormously the fit, making it much faster and lighter. These light fits, together with the new methodology, allow us to explore an space of parameters previously inaccessible.

## 2.5 Performance benchmark

In order to obtain a good quality and reliable PDF model it is necessary to perform the fit for many artificial data replicas. These are complex computation which require a great deal of CPU hours and memory consumption, therefore one of the goals of any new studies is to find a more efficient way of performing the PDF fits. As previously stated, GD methods improve the stability of the fits, producing less “bad replicas”, which need to be discarded, than their GA counterparts and this translates to a much smaller computing time. In Table 1 we find a factor of 20 improvement with respect to the old methodology and near to a factor of 1.5 in the percentage of accepted replicas for a global fit setup.

In the old code the memory usage is driven by the usage of APFEL [21], which does not depend on the set of experiments being used. Instead, the memory consumption of the new code is driven by the Tensorflow optimization strategy which in the case of hadronic data requires the implementation of Eq. (4) and its gradient. This difference translates to an importance increase on the memory usage of `n3fit` that is only realized in the Global fit.

We are currently working on ways that would allow us to reduce the memory consumption without introducing a penalty on the execution speed of the code as currently we favour speed with respect to memory.

## 3 Hyperparameter tuning

The NNPDF approach aimed to reduce the bias introduced in the determination of the functional form utilized to parametrize the PDFs [22]. Neural networks provide universal function approximators [23] which reduce systematic biases introduced by the choice of specific functional forms. Neural networks themselves, however, are not unique and the space of hyperparameters is big enough that finding the best choice becomes a overwhelming task.

In this work we aim to improve over the previous iteration of the NNPDF methodology by boxing the entire framework under hyperparameter optimization routines, there are several key points which allow us to do this now. Firstly, the new design of the code exposes all parameters of the fit including (but not restricted to) the neural network architecture. This is of key importance for a proper hyperparameter scan where everything is potentially interconnected. Secondly, the new methodology has such a smaller impact on computing resources that we can perform more fits on a scale of orders of magnitude, in other words, for each fit using the old methodology we can now test hundreds of architectures.

The hyperparameter scan capabilities are implemented using the hyperopt framework [17] which systematically scans over a selection of parameter using Bayesian optimiza-

**Table 1** Comparison of the average computing resources consumed by the old and new methodologies for the DIS and Global setups. We find `n3fit` to be  $\sim 20$  faster on average. The only drawback is the big-

DIS fit	CPU h.	Mem. usage (GB)	Good replicas (%)
<code>n3fit</code> (new)	0.2	2	95
<code>nnfit</code> (old)	4	4	70
Global fit	CPU h.	Mem. usage (GB)	Good replicas (%)
<code>n3fit</code> (new)	1.5	14	95
<code>nnfit</code> (old)	30	5	70

**Table 2** Parameters on which the hyperparameter scan is performed. Results marked with (\*) are shown graphically in Fig. 3

Neural network	Fit options
Number of layers (*)	Optimizer (*)
Size of each layer	Initial learning rate (*)
Dropout	Maximum number of epochs (*)
Activation functions (*)	Stopping Patience (*)
Initialization functions (*)	Positivity multiplier (*)

tion [24] and measures model performance to select the best architecture.

As a proof of concept, for this paper we make a first selection of parameters on which to scan, shown in Table 2 separated between the parameters which define the Neural Network architecture and those which define the fitting procedure.

In this study we apply the framework to both the global and DIS only setup and in order to achieve the best model configuration we limit the data input to the experimental central values instead of using artificial replicas. We optimize on a combination of the best validation loss and stability of the

ger memory consumption in the global fit. Each fit can be comprised of 100–200 replicas. Good replicas are those which pass all post-fit criteria defined in [8]

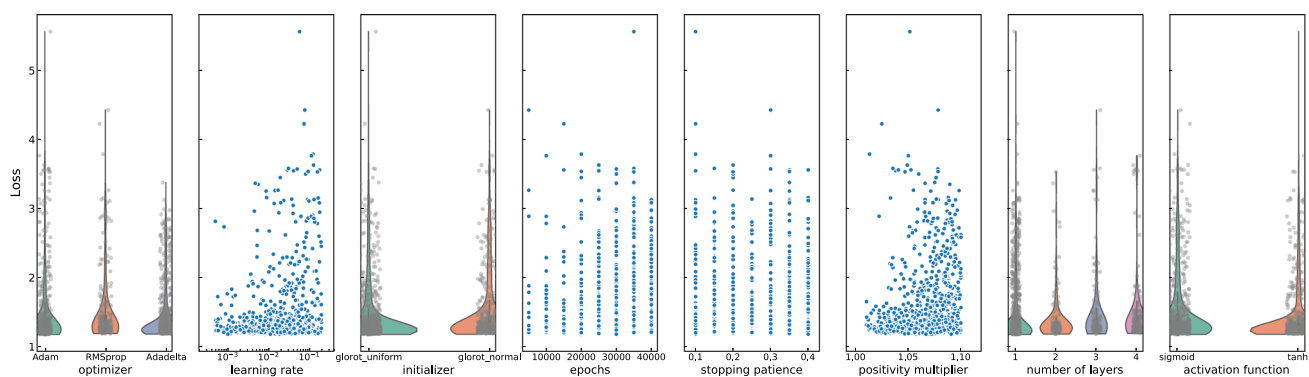
fits. In other words, we select the architecture which produces the lowest validation loss after we trim those combinations which are deemed to be unstable.

In Fig. 3 we show an example of DIS only scan. We present eight examples of those shown in Table 2.

In this scan we find the Adadelta optimizer, for which no learning rate is used, to be more stable and systematically produce better results than RMSprop and Adam with a wide choice of learning rates. The initializers, once unstable options such as a random uniform initialization have been removed, seem to provide similar qualities with a slight preference for the “glorot\_normal” initialization procedure described in [25].

Concerning the parameters related to the stopping criteria, we observe that when the number of epochs is very small the fit can be certainly unstable, however after a certain threshold no big differences are observed. The stopping patience shows a very similar pattern, stopping too early can be disadvantageous but stopping too late does not seem to make a big difference. The positivity multiplier, however, shows a clear preference for bigger numbers.

Finally, concerning the neural network architecture we observe that a small number of layers seem to produce



**Fig. 3** Graphical representation of an hyperparameter scan for a DIS only fit with 2000 trials. The loss presented in the y-axis corresponds to an average of the validation and testing loss functions. The shape of

the violin plots represent a visual aid on the behaviour of the fit as a function of the free parameter. Fatter plots represent better stability, i.e., configurations which are less likely to produce outliers



slightly better absolute results, however, one single hidden layer seem to be also very inconsistent. The activation functions present with a slight preference for the hyperbolic tangent. Once we have a acceptable hyperparameter setup we ran again for fine tuning as some of the choices could have been biased by a bad combination on the other parameters.

The main take away from this scan is the implementation of a semi-automatic methodology able to find the best hyperparameter combination as the setup changes, e.g. with new experimental data, new algorithms or technologies.

### 3.1 Overfitting: the test set

While performing the hyperparameter scan we found that optimizing only looking at the validation loss produced results which would usually be considered overfitted: very low training and validation  $\chi^2$  and complex replica patterns. Thanks to the high performance of the `n3fit` procedure the usual cross-validation algorithm used in the NNPDF framework was not enough to prevent overlearning for all architectures.

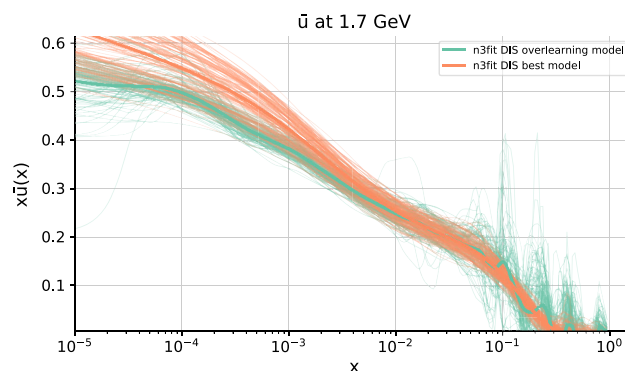
The cross-validation implemented in NNPDF is successful on avoiding the learning of the noise within a dataset. However, we observe that this choice is not enough to prevent overfitting due to correlations within points in a same dataset when using hyperopt with `n3fit`. In order to eliminate architectures that allowed overlearning we proceed by including a testing set where the model generalization power is tested. This is a set of datasets where none of the points are used in the fitting either for training or validation.

Defining the best appropriate test dataset for PDF fits is particularly challenging due to the nature of the model regression through convolutions. For the present results the test set is defined by removing from the training set datasets with duplicate process type and smaller leading-order kinematic range coverage. The choice of a particular prescription for the test set presents a certain level of arbitrariness. In this study we focus our attention to the prediction quality at the level of physical observables. We believe that more sophisticated prescriptions based on the constraining power of data for specific PDF flavours and momentum fraction regions, in combination with the overall observable quality, may provide a stronger definition for the test set. We call the loss produced by the removed datasets “testing loss” and we use it as a third criterion (beyond stability and combined with the value of the validation loss) to discard combinations of hyperparameters. With this procedure we are able to find combinations of hyperparameters which produce good fits for which we are confident no obvious overfitting can be generated. In Table 3 we list the datasets which have been used as test set for this study.

In Fig. 4 we show an example of a PDF produced by two very different architectures, both of which are generated by

**Table 3** List of datasets used for the testing procedure of the hyperparameter scan. After each fit the generalization power of the network is tested on these sets and the iteration discarded if its  $\chi^2$  greatly differs from the validation and training  $\chi^2$ . This list is a subset of the datasets entering the fits for NNPDF 3.1 [4]

Experiment	Observable
NMC	$\sigma_{\text{NC}}^e$
BCDMS	$F_2^p, F_2^d$
HERA	$\sigma_{\text{NC}}^p$ (460 GeV), $F_2^b$
D0	$\frac{d\sigma_Z}{dy_z}$ , W electron asym
CDF	$k_t$ incl jets
ATLAS	Mass DY, $\sigma(t\bar{t})$
CMS	$Z p_T$ 8 TeV



**Fig. 4** Comparison between the PDF replicas generated by `n3fit` for one parton flavour ( $\bar{u}$ ). In green we show the results for the best model in the naive hyperoptimization, in orange the best model once we have introduced the test set criteria

the hyperoptimization procedure. We observe a much more unstable behaviour in the fit in which we allow for overtraining which in turn translates for a  $\chi^2$  on the testing set of more than twice the value of the training set. We believe the issue of hidden correlations in experimental measurements as well as its impact on PDF fits requires a much deeper study outside the scope of this paper.

## 4 Results

The best setups we find are shown in Table 4. We find the global setup allow for deeper networks without falling in overfitting. The hyperbolic tangent and the sigmoid functions are found to perform similarly. The initializer of the weights of the network, however, carries some importance for the stability of the fits. We utilize the glorot normal initialization method [25,26] as implemented in Keras.

We find that adding a small dropout rate [27] to the hidden layers in the global fit reduces the chance of overlearning introduced by the deeper network, thus achieving more stable results. As expected the bigger network shows a certain

**Table 4** Best models found by our hyperparameter scan for the DIS and global setups using the new `n3fit` methodology

Parameter	DIS only	Global
Hidden layers	2	3
Architecture	35-25-8	50-35-25-8
Activation	tanh	sigmoid
Initializer	glorot_normal	glorot_normal
Dropout	0.0	0.006
Optimizer	Adadelta	Adadelta
Max epochs	40,000	50,000
Stopping patience	30%	30%

**Table 5** Comparison of the total  $\chi^2$  of the fit for both a DIS only and Global setup between the old and new methodology

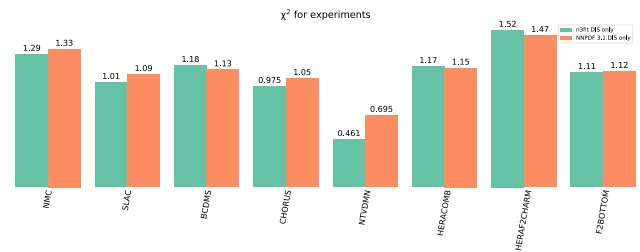
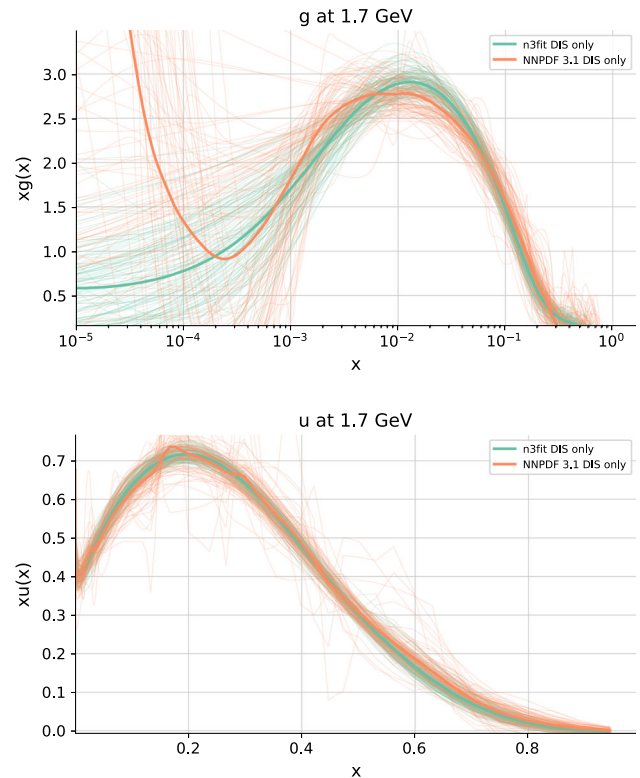
	DIS only	Global
<code>n3fit</code> (new)	1.10	1.15
<code>nnfit</code> (old)	1.13	1.16

preference for greater waiting times (which also increases the stopping patience as is set to be a % of the maximum number of epochs). In reality the max. number of epochs is rarely reached and very few replicas are wasted.

We have produced two complete fits for the DIS only and global setups described in Sect. 2.4. We find that both fits perform similarly on describing the experimental data, as can be attested by the values of  $\chi^2$  presented in Table 5. Below we detail the results separating between the DIS only and global setups and showing a direct comparison between the behaviour of the PDFs found with both methodologies. In all plots of this section the orange color corresponds to the fit performed with the old methodology whereas green corresponds to the new one and are generated using `reportengine` [28].

#### 4.1 DIS only fits

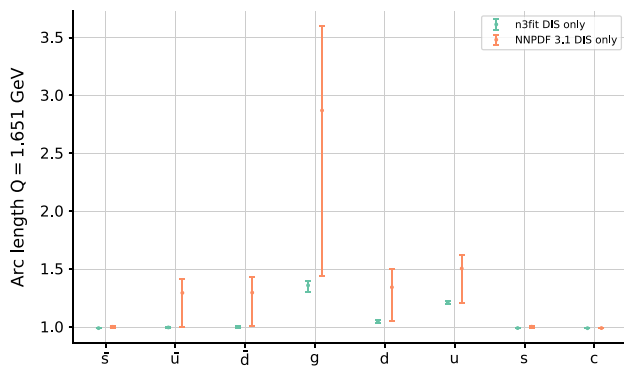
When we study the change on the value of  $\chi^2$  in Table 5 it is interesting to study also how this value changes experiment by experiment. In Fig. 5 we compare the individual  $\chi^2$  experiment by experiment between the new and old methodology. We observe that values are compatible within the statistical fluctuations obtained by changing the random seed during the initialization of the old methodology. From this we can infer the behaviour of the PDFs must also be similar. We show some examples for the gluon and  $u$ -quark replicas in Fig. 6. Indeed, the central value for the PDF is not very different from the one obtained with the old methodology (albeit somewhat smoother) and always lying within the one sigma band of the old PDF fits.

**Fig. 5** Comparison of the  $\chi^2$  experiment by experiment between the new and old codes. An experiment is comprised of one or more datasets of experimental data. All datasets in this plot correspond to DIS-type experiments**Fig. 6** Comparison of the PDF of the gluon and the  $u$ -quark for the DIS only setup. Each line correspond to a different replica while the bold-faced line correspond to the central value of the PDF computed by taking the average of all other replicas

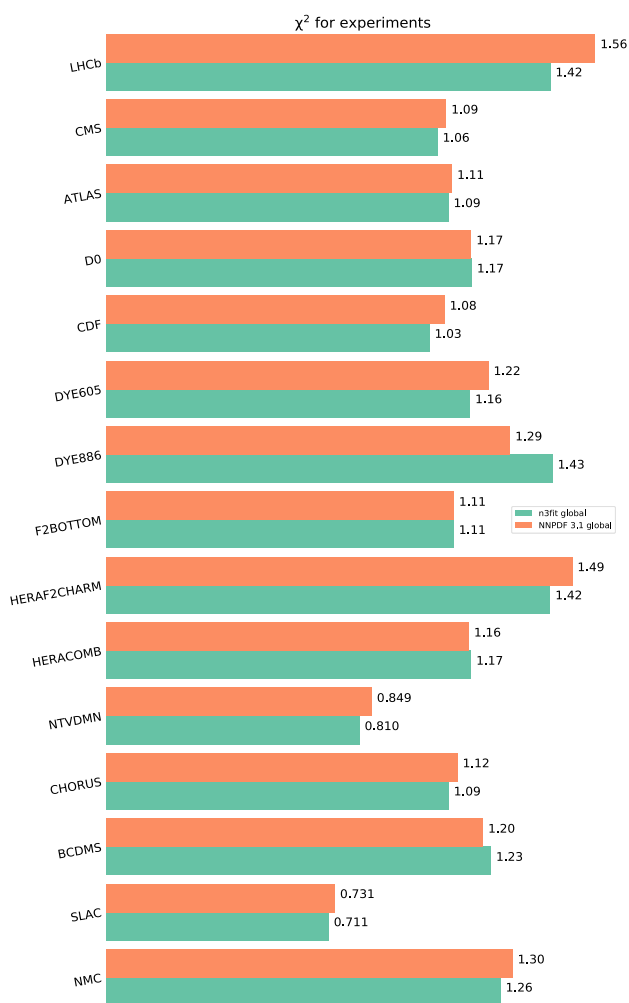
The biggest difference between both methodologies resides on the stability of the replicas. In Fig. 6 we can see that `n3fit` produces smoother replicas with less complex structure. The central value, however, remains stable and well within the envelope of NNPDF3.1. As a result in Fig. 7 we observe that the arc-lengths for all flavours are systematically smaller with `n3fit`.

#### 4.2 Global fits

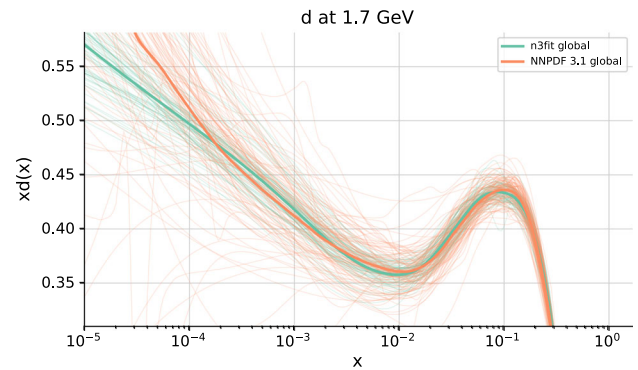
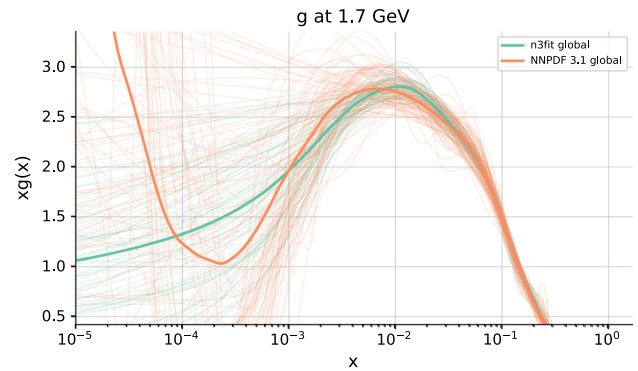
Similar results are obtained in the case of the global fit. The  $\chi^2$  per experiment is shown in Fig. 8. Although the differ-



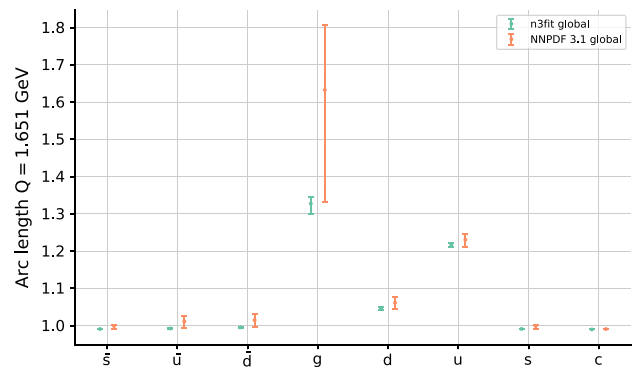
**Fig. 7** Comparison for the DIS only setup of the arc-lengths between the new and old methodologies. The new methodology produces smoother curves which translates to smaller and more stable arc-lengths



**Fig. 8** Comparison of the  $\chi^2$  between both the new and old codes experiment by experiment for a global fit. An experiment is comprised of one or more datasets of experimental data. We find the new methodology to be able to produce fits with a quality similar to the old methodology for every experiment



**Fig. 9** Comparison of the PDF of the gluon and the  $d$ -quark for the global setup. Each line correspond to a different replica while the bold-faced line correspond to the central value of the PDF computed by taking the average of all other replicas



**Fig. 10** Comparison for the global setup of the arc-lengths between the new and old methodologies. As in the DIS setup, the new methodology produces smoother curves with smaller and more stable arc-lengths

ence between the old and new methodology in the global setup is not as evident as in the DIS only case, we can still observe more stable replicas and in general smoother curves in Fig. 9, where we plot all produced replicas for the new and old methodologies for the gluon and  $d$ -quark PDFs.

The same is observed in Fig. 10 where the arc-lengths produced by `n3fit` are still systematically smaller. It is also worth noticing the more stable behaviour of `n3fit` with respect to NNPDF3.1 when comparing the DIS only arc-lengths of Fig. 7 with those produced by the global fit.



This difference can be easily understood as we now possess a framework able to scan and search for the best combination of hyperparameters for a given experimental setup which allow us to obtain good quality fits for any setup using the same framework. Using the old methodology a similar study would have required several months of work. This is another example of how this new methodology can improve the field of PDF determination.

## 5 Conclusions

We presented a new approach to and regression model strategy for the determination of PDFs, in the context of the NNPDF framework. This new approach, based on new computational techniques, improves fitting performance and quality. Furthermore, we propose a new workflow pipeline for the systematic and efficient determination of PDFs.

The new approach consists in replacing the current C++ fitting code with a new implementation based on python and Keras-Tensorflow. This allows us to change model easily, test new architectures developed by the scientific community, fit preprocessing exponents, obtain faster results thanks to gradient based methods, and the possibility to carry hyperparameter tuning in a systematic way to decide when the model is optimal.

Finally, we believe that future PDF fits based on this new approach, once exhaustively tested and validated, will provide better and more reliable PDF sets for future releases of the NNPDF collaboration, thanks to the possibility to identify in a quantitative way the best suited regression model for new data from the LHC.

**Acknowledgements** We thank Stefano Forte, Juan Rojo and Jake Either for careful reading of the manuscript. We also thank the members of the NNPDF collaboration for useful and insightful discussions. This project is supported by the European Research Council under the European Unions Horizon 2020 research and innovation Programme (Grant Agreement Number 740006).

**Data Availability Statement** This manuscript has no associated data or the data will not be deposited. [Authors' comment: This work presents conceptual developments therefore no data is available.]

**Open Access** This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made. Funded by SCOAP<sup>3</sup>.

## References

1. J. Butterworth et al., J. Phys. G **43**, 023001 (2016). [arXiv:1510.03865](https://arxiv.org/abs/1510.03865)
2. K. Kovarik, P.M. Nadolsky, D.E. Soper, (2019). [arXiv:1905.06957](https://arxiv.org/abs/1905.06957)
3. R. Abdul Khalek, S. Bailey, J. Gao, L. Harland-Lang, J. Rojo, (2019). [arXiv:1906.10127](https://arxiv.org/abs/1906.10127)
4. R.D. Ball et al. (NNPDF), Eur. Phys. J. C **77**, 663 (2017). [arXiv:1706.00428](https://arxiv.org/abs/1706.00428)
5. V. Bertone, S. Carrazza, N.P. Hartland, E.R. Nocera, J. Rojo (NNPDF), Eur. Phys. J. C **77**, 516 (2017). [arXiv:1706.07049](https://arxiv.org/abs/1706.07049)
6. S. Carrazza, N.P. Hartland, J. Phys. Conf. Ser. **1085**, 052007 (2018). [arXiv:1711.09991](https://arxiv.org/abs/1711.09991)
7. V. Bertone, N.P. Hartland, E.R. Nocera, J. Rojo, L. Rottoli (NNPDF), Eur. Phys. J. C **78**, 651 (2018). [arXiv:1807.03310](https://arxiv.org/abs/1807.03310)
8. R.D. Ball et al. (NNPDF), JHEP **04**, 040 (2015). [arXiv:1410.8849](https://arxiv.org/abs/1410.8849)
9. R.D. Ball, L. Del Debbio, S. Forte, A. Guffanti, J.I. Latorre, J. Rojo, M. Ubiali, Nucl. Phys. B **838**, 136 (2010). [arXiv:1002.4407](https://arxiv.org/abs/1002.4407)
10. V. Bertone, S. Carrazza, N.P. Hartland, Comput. Phys. Commun. **212**, 205 (2017). [arXiv:1605.02070](https://arxiv.org/abs/1605.02070)
11. R.D. Ball et al., JHEP **04**, 125 (2013). [arXiv:1211.5142](https://arxiv.org/abs/1211.5142)
12. R. Abdul Khalek et al., (2019). [arXiv:1905.04311](https://arxiv.org/abs/1905.04311)
13. F. Chollet et al., Keras (2015). <https://keras.io>
14. M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G.S. Corrado, A. Davis, J. Dean, M. Devin et al., TensorFlow: large-scale machine learning on heterogeneous systems (2015). <http://tensorflow.org/>
15. B.T. Wang, T.J. Hobbs, S. Doyle, J. Gao, T.J. Hou, P.M. Nadolsky, F.I. Olness, Phys. Rev. D **98**, 094030 (2018). [arXiv:1803.02777](https://arxiv.org/abs/1803.02777)
16. R. Abdul Khalek, J.J. Ethier, J. Rojo (NNPDF), Eur. Phys. J. C **79**, 471 (2019). [arXiv:1904.00018](https://arxiv.org/abs/1904.00018)
17. J. Bergstra, D. Yamins, D.D. Cox, Making a science of model search: hyperparameter optimization in hundreds of dimensions for vision architectures, in *Proceedings of the 30th International Conference on International Conference on Machine Learning, ICML'13*, vol. 28, pp. I-115–I-123 (JMLR.org, 2013). <http://dl.acm.org/citation.cfm?id=3042817.3042832>
18. M.D. Zeiler, CoRR (2012). [arXiv:1212.5701](https://arxiv.org/abs/1212.5701)
19. D. Kingma, J. Ba, in *International Conference on Learning Representations* (2014)
20. T. Tieleman, G. Hinton, Lecture 6.5—RmsProp: divide the gradient by a running average of its recent magnitude, in *COURSERA: Neural Networks for Machine Learning* (2012)
21. V. Bertone, S. Carrazza, J. Rojo, Comput. Phys. Commun. **185**, 1647 (2014). [arXiv:1310.1394](https://arxiv.org/abs/1310.1394)
22. S. Forte, L. Garrido, J.I. Latorre, A. Piccione, JHEP **05**, 062 (2002). [arXiv:hep-ph/0204232](https://arxiv.org/abs/hep-ph/0204232)
23. G. Cybenko, Math. Control Signals Syst. **2**, 303 (1989)
24. J. Bergstra, R. Bardenet, Y. Bengio, B. Kégl, *Algorithms for hyper-parameter optimization*, in *Proceedings of the 24th International Conference on Neural Information Processing Systems* (Curran Associates Inc., USA, 2011), NIPS'11, pp. 2546–2554, ISBN 978-1-61839-599-3. <http://dl.acm.org/citation.cfm?id=2986459.2986743>
25. X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in *Proceedings of the International Conference on Artificial Intelligence and Statistics (AISTATS'10)*. Society for Artificial Intelligence and Statistics (2010)
26. Y. Bengio, X. Glorot, in *International Conference on Artificial Intelligence and Statistics* (2010), pp. 249–256
27. G.E. Hinton, N. Srivastava, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, CoRR (2012). [arXiv:1207.0580](https://arxiv.org/abs/1207.0580)
28. Z. Kassabov, Reportengine: a framework for declarative data analysis (2019). <https://doi.org/10.5281/zenodo.2571601>